



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/602,551	06/24/2003	Thomas A. Makowski	5150-80201	1235

7590 06/22/2009  
Jeffrey C. Hood  
Meyertons, Hood, Kivlin,  
Kowert & Goetzel PC  
P.O. Box 398  
Austin, TX 78767

EXAMINER
----------

DAO, THUY CHAN

ART UNIT	PAPER NUMBER
----------	--------------

2192

MAIL DATE	DELIVERY MODE
-----------	---------------

06/22/2009

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

---

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/602,551  
Filing Date: June 24, 2003  
Appellant(s): MAKOWSKI ET AL.

---

Jeffrey C. Hood  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed August 5, 2008 appealing from the Office action mailed March 10, 2008, the reply brief filed December 12, 2008, and the order returning undocketed appeal to examiner on April 27, 2009.

In view of the order returning undocketed appeal to examiner on April 27, 2009, please note the actions on this instant Examiner's Answer as follows:

- The Examiner's Answer mailed on October 28, 2008 has been vacated.
- In the instant Examiner's Answer, a typographical error on the Ground of rejection has been corrected, i.e.,

Examiner notes that Kudukoli (US Patent Publication No. 2001/0024211 A1) was previously applied under 35 USC **102(b)** rejection in the Office actions mailed April 19, 2007 (page 3); October 16, 2007 (page 3), and March 10, 2008 at page 3, item 7. However, under item 8, a typographical error was listed as "**102(e)**" instead.

Accordingly, the correct ground of rejection is now set forth: Claims 69-92 are rejected under 35 U.S.C. **102(b)** as being anticipated by Kudukoli (US Patent Publication No. 2001/0024211 A1).

- The Reply Brief filed December 12, 2008 has been entered and considered.
- Previously mailed Examiner's Answer is reproduced and re-applied herein with the correct Ground of rejection **102(b)**.

#### **(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

#### **(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

#### **(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

#### **(4) Status of Amendments After Final**

Art Unit: 2192

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

2001/0024211 A1

KUDUKOLI et al.

09-2001

-O-O-O-

**(9) Grounds of Rejection**

☐ The following grounds of rejection are applicable to the appealed claims:

Claims 69-92 are rejected under 35 U.S.C. 102(b) as being anticipated by Kudukoli (art of record, US Patent Publication No. 2001/0024211 A1).

**Claim 69:**

Kudukoli discloses *a computer-accessible memory medium that stores program instructions executable by a processor to perform:*

*displaying a node in a graphical program (e.g., FIG. 13, displaying a new VI Object Reference Node in a VI graphical program, [0212-0221]);*

*receiving first user input invoking display of a plurality of functions for the node; displaying the plurality of functions for the node in response to the first user input; (e.g., FIG. 21, from said new VI Object Node, invoking display of a plurality of functions such as "Numeric", "Digital", "Slide", ..., [0215-0216]; FIG. 25, section 2, [0278], from said new VI Object Node, invoking display of functions "Control", "User Control", "ActiveX" ..., see also FIG. 22);*

*receiving second user input selecting a function from the plurality of functions (e.g., FIG. 22, from "Slide", selecting a specific function "Vertical Pointer Slide", [0217]; from "User Control", selecting a specific function "Waveform Chart", see also [0278] and FIG. 22);*

*determining graphical program code based on the second user input (e.g., FIG. 4, blocks 208-210; [0022], [0029]; [0034]),*

*wherein the determined graphical program code comprises a graphical representation of an implementation of the selected function (e.g., FIG. 6, blocks 302-308, [0136-0140]; FIG. 4, [0100-0113]; FIG. 25A, Section 2; FIG. 25C, Section 8, [0281]; [0129]), and*

*wherein the determined graphical program code is executable to provide functionality in accordance with the selected function; associating the determined graphical program code with the node (e.g., FIG. 4, [0100-0113]; FIG. 23, [0273-0275]; FIG. 25A-C, [0120]; [0022]),*

Art Unit: 2192

*wherein, when the node in the graphical program executes, the determined graphical program code executes to provide the functionality in accordance with the selected function (e.g., FIG. 6, [0136-0140]; FIG. 4, blocks 208-210, [0273-0275]; [0029]; FIG. 25A-C).*

**Claim 70:**

The rejection of claim 69 is incorporated. Kudukoli also discloses:

*the node has a first node icon which is displayed in the graphical program, and wherein the first node icon has a first appearance (e.g., [0130], [0269]),*

*wherein the program instructions are further executable to perform: changing the first node icon to a second appearance based on the second user input (e.g., [0030], [0130]),*

*wherein said changing the first node icon to a second appearance includes displaying an image corresponding to the selected function (e.g., [0269]).*

**Claim 71:**

The rejection of claim 70 is incorporated. Kudukoli also discloses *said changing the first node icon to a second appearance comprises replacing the first node icon with a second node icon (e.g., [0225], [0230]).*

**Claim 72:**

The rejection of claim 69 is incorporated. Kudukoli also discloses *prior to said associating the determined graphical program code with the node, the node does not have any associated graphical program code (e.g., [0212-0221]).*

**Claim 73:**

The rejection of claim 69 is incorporated. Kudukoli also discloses:

*prior to said associating the determined graphical program code with the node, the node has associated default graphical program code in accordance with a default function for the node (e.g., [0225]), and*

Art Unit: 2192

*wherein the default graphical program code implements a first functionality; and wherein said associating the determined graphical program code with the node comprises replacing the default graphical program code with the determined graphical program code (e.g., [0230]).*

**Claim 74:**

The rejection of claim 69 is incorporated. Kudukoli also discloses *said receiving first user input comprises receiving the first user input to the node; and wherein said receiving second user input comprises receiving the second user input to the node (e.g., FIG. 21 and related text).*

**Claim 75:**

The rejection of claim 69 is incorporated. Kudukoli also discloses:

*said displaying the plurality of functions for the node in response to the first user input comprises: displaying a plurality of function classes for the node (e.g., FIG. 25, section 2, [0278]); and*

*in response to user input selecting a function class, displaying the plurality of functions, wherein the plurality of functions are in the selected function class (e.g., [0217], [0278]).*

**Claim 76:**

The rejection of claim 69 is incorporated. Kudukoli also discloses:

*the node is a data acquisition (DAQ) node (e.g., [0015], [0085]);*

*wherein the plurality of functions for the node comprise a plurality of DAQ functions (e.g., [0080], [0092]);*

*wherein, prior to said associating, the DAQ node comprises one of: a generic read node; a generic write node; a generic channel creation node; a generic timing node; or a generic triggering node (e.g., [0085]); and*

*wherein, after said associating, the DAQ node comprises one of: a specific read node in accordance with the selected function; a specific write node in accordance*

Art Unit: 2192

*with the selected function; a specific channel creation node in accordance with the selected function; a specific timing node in accordance with the selected function; or a specific triggering node in accordance with the selected function (e.g., [0092], [0015]).*

**Claim 77:**

Kudukoli also discloses *a computer-implemented method for configuring a graphical program node, comprising:*

*displaying a node in a graphical program (e.g., FIG. 13, [0212-0221]);*

*receiving first user input invoking display of a plurality of functions for the node; displaying the plurality of functions for the node in response to the first user input (e.g., FIG. 21, [015-0216]; FIG. 25, [0278] and FIG. 22);*

*receiving second user input selecting a function from the plurality of functions (e.g., [0022], [0029], [0034]);*

*determining graphical program code based on the second user input (e.g., FIG. 6, [0136-0140]; [0100-0130]),*

*wherein the determined graphical program code comprises a graphical representation of an implementation of the selected function (e.g., [0022], [0029], [0034]), and*

*wherein the determined graphical program code is executable to provide functionality in accordance with the selected function (e.g., [0136-0140]; [0281], [0129]);*

*associating the determined graphical program code with the node, wherein, when the node in the graphical program executes (e.g., FIG. 4, [0100-0113]; FIG. 23, [0273-0275]),*

*the determined graphical program code executes to provide the functionality in accordance with the selected function (e.g., [0029], [0136-0140]).*

**Claim 78:**

The rejection of claim 77 is incorporated. Kudukoli also discloses:



*the node has a first node icon which is displayed in the graphical program, the method further comprising: changing the first node icon to a second appearance based on the second user input (e.g., [0130], [0269]),*

*wherein said changing the first node icon to a second appearance includes displaying an image corresponding to the selected function (e.g., [0225], [0230]).*

**Claim 79:**

The rejection of claim 78 is incorporated. Kudukoli also discloses *said changing the first node icon to a second appearance comprises replacing the first node icon with a second node icon (e.g., [0225]).*

**Claim 80:**

The rejection of claim 77 is incorporated. Kudukoli also discloses: *prior to said associating the determined graphical program code with the node, the node does not have any associated graphical program code (e.g., [0230]).*

**Claim 81:**

The rejection of claim 77 is incorporated. Kudukoli also discloses:  
*prior to said associating the determined graphical program code with the node, the node has associated default graphical program code in accordance with a default function for the node (e.g., [0130], [0225]), and*  
*wherein the default graphical program code implements a first functionality; and wherein said associating the determined graphical program code with the node comprises replacing the default graphical program code with the determined graphical program code (e.g., [0269], [0230]).*

**Claim 82:**

The rejection of claim 77 is incorporated. Kudukoli also discloses *said receiving first user input comprises receiving the first user input to the node; and wherein said*

Art Unit: 2192

*receiving second user input comprises receiving the second user input to the node (e.g., [0022], [0034]).*

**Claim 83:**

The rejection of claim 77 is incorporated. Kudukoli also discloses *said displaying the plurality of functions for the node in response to the first user input comprises: displaying a plurality of function classes for the node; and in response to user input selecting a function class, displaying the plurality of functions, wherein the plurality of functions are in the selected function class (e.g., [0215-0216]; [0278]).*

**Claim 84:**

The rejection of claim 77 is incorporated. Kudukoli also discloses  
*the node is a data acquisition (DAQ) node (e.g., [0015], [0085]);*  
*wherein the plurality of functions for the node comprise a plurality of DAQ functions (e.g., [0080], [0092]);*  
*wherein, prior to said associating, the DAQ node comprises one of: a generic read node; a generic write node; a generic channel creation node; a generic timing node; or a generic triggering node (e.g., [0085]); and*  
*wherein, after said associating, the DAQ node comprises one of: a specific read node in accordance with the selected function; a specific write node in accordance with the selected function; a specific channel creation node in accordance with the selected function; a specific timing node in accordance with the selected function; or a specific triggering node in accordance with the selected function (e.g., [0092], [0015]).*

**Claim 85:**

Kudukoli also discloses *a computer-accessible memory medium that stores program instructions executable by a processor to perform:*  
*displaying a node in a graphical program (e.g., FIG. 13, [0212-0221]);*

*receiving first user input invoking display of a plurality of functions for the node; displaying the plurality of functions for the node in response to the first user input (e.g., FIG. 21, [0215-0216]; FIG. 25, [0278]);*

*receiving second user input selecting a function from the plurality of functions (e.g., FIG. 22, [0217]);*

*determining a second node based on the selected function, wherein the second node comprises a graphical representation of an implementation of the selected function (e.g., [0022], [0029], [0034]), and*

*wherein the second node comprises graphical program code executable to provide functionality in accordance with the selected function (e.g., [0136-0140], [0100-0113]);*

*replacing the node in the graphical program with the second node (e.g., [0225]; [0230], FIG. 22 “Vertical Pointer Slide” as the second node, and FIG. 4, [0100-0113]; FIG. 25A-C),*

*wherein, when the second node in the graphical program executes, the graphical program code of the second node executes to provide the functionality in accordance with the selected function (e.g., FIG. 25A, Section 2 and FIG. 25C, Section 8, [0136-0140], [0029]).*

**Claim 86:**

The rejection of claim 85 is incorporated. Kudukoli also discloses *the node comprises a first node icon, and wherein said displaying the node comprises displaying the first node icon, and wherein the second node comprises: the first node icon and the graphical program code; or a second node icon and the graphical program code (e.g., [0215-0216]).*

**Claim 87:**

The rejection of claim 85 is incorporated. Kudukoli also discloses *the node and/or the second node is one or more of: polymorphic; function switchable; or function class switchable (e.g., [0217] and [0278]).*

**Claim 88:**

The rejection of claim 85 is incorporated. Kudukoli also discloses:

*the node is a data acquisition (DAQ) node (e.g., [0015], [0085]);*

*wherein the DAQ node comprises one of: a generic read node; a generic write node; a generic channel creation node; a generic timing node; or a generic triggering node (e.g., [0080], [0092]); and*

*wherein the second node comprises a corresponding one of: a specific read node in accordance with the selected function; a specific write node in accordance with the selected function; a specific channel creation node in accordance with the selected function; a specific timing node in accordance with the selected function; or a specific triggering node in accordance with the selected function (e.g., [0092], [0015]).*

**Claim 89:**

Kudukoli also discloses a computer-implemented method for configuring a graphical program node, comprising:

*displaying a node in a graphical program (e.g., FIG. 13, [0212-0221]);*

*receiving first user input invoking display of a plurality of functions for the node; displaying the plurality of functions for the node in response to the first user input (e.g., [0215-0216]);*

*receiving second user input selecting a function from the plurality of functions (e.g., [0217], [0278]);*

*determining a second node based on the selected function, wherein the second node comprises a graphical representation of an implementation of the selected function (e.g., [0022], [0029], [0034]), and*

*wherein the second node comprises graphical program code executable to provide functionality in accordance with the selected function (e.g., [0136-0140]);*

*replacing the node in the graphical program with the second node (e.g., [0100-0113]),*

*wherein, when the second node in the graphical program executes, the determined graphical program code executes to provide the functionality in accordance with the selected function (e.g., [0136-0140], [0225], [0230]).*

**Claim 90:**

The rejection of claim 89 is incorporated. Kudukoli also discloses *the node comprises a first node icon, and wherein said displaying the node comprises displaying the first node icon, and wherein the second node comprises: the first node icon and the graphical program code; or a second node icon and the graphical program code (e.g., FIG. 21 and related text, [0217]).*

**Claim 91:**

The rejection of claim 89 is incorporated. Kudukoli also discloses *the node and/or the second node is one or more of: polymorphic; function switchable; or function class switchable (e.g., [0278]).*

**Claim 92:**

The rejection of claim 89 is incorporated. Kudukoli also discloses:

*the node is a data acquisition (DAQ) node (e.g., [0015], [0085]);*

*wherein the DAQ node comprises one of: a generic read node; a generic write node; a generic channel creation node; a generic timing node; or a generic triggering node (e.g., [0085]); and*

*wherein the second node comprises a corresponding one of: a specific read node in accordance with the selected function; a specific write node in accordance with the selected function; a specific channel creation node in accordance with the selected function; a specific timing node in accordance with the selected function; or a specific triggering node in accordance with the selected function (e.g., [0092], [0015]).*

**(10) Response to Argument**

Claims 69-92 were rejected under 35 U.S.C. 102(b) as being anticipated by Kudukoli (US Pat. Pub. No. 2001/0024211 A1).

**Independent Claims 69 and 77 (Brief, pp. 6-9):**

The limitations at issue “associating the determined graphical program code with the node, wherein, when the node in the graphical program executes, the determined graphical program code executes to provide the functionality in accordance with the selected function” (e.g., independent claim 69, lines 12-14 and similarly recited in independent claim 77, lines 12-14, emphasis added).

At the outset, the examiner notes wherein the ground of rejections is based on figures and text extracted from the reference Kudukoli, the Appellant merely provided general assertions regarding the reference Kudukoli without providing proof nor pointing out the evidence cited from the reference Kudukoli (e.g., Brief, page 7, lines 4-20, emphasis added).

In Brief, page 7, line 28 – page 8, line 1, the Appellant further stated,

“Thus, Appellant submits that even if Kudukoli's New VI Object Reference node code that generates a new VI object for new graphical program were considered to be Appellant's claimed determined graphical program code, ... thus no associating of this code with the node is performed.” (emphasis added).

The examiner respectfully submits that the Appellant mischaracterized Kudukoli's teachings.

Kudukoli teaches after the New VI Object Reference Node (FIG. 13, [0212]-[0217]) receives first user input (FIG. 21, [0215], selecting a “vi object class” such as selecting “slide”) and second user input (FIG. 22, [0217], selecting a “style”/“sub-class” of the “vi object class” such as selecting “vertical pointer slide”), said New VI Object

Reference Node becomes/is replaced by a Vertical Pointer Slide node ([0213] and [0217], emphasis added), and that is not "...New VI Object Reference node code that generates a new VI object" as Appellant contended without providing proof and/or evidence.

Claim 69	Kudukoli
<i>displaying a node in a graphical program;</i> <i>receiving first user input...;</i>  <i>receiving second user... ,</i>  <i>wherein the determined graphical program code comprises a graphical representation of an implementation of the selected function, and</i>   <i>... associating the determined graphical program code with the node,</i>  <i>wherein, when the node in the graphical program executes,</i>  <i>the determined graphical program code executes to provide the functionality in accordance with the selected function.</i>	displaying a New VI Object Reference Node (FIG. 13, [0212]-[0217]); New VI Obj. Ref. Node receives first user input (FIG. 13, [0215]) New VI Obj. Ref. Node receives second user input (FIG. 13, [0217])  FIG. 4, blocks 208-210  New VI Obj. Ref. Node becomes/ is replaced by a "vertical pointer slide node" ([0213] and [0217]); a "waveform chart UI node" (FIG. 25A, Section 2, [0278]); or "a wait function node" (FIG. 25C, Section 8, [0284])  ... associating code with said above nodes indeed means associating code with the New VI Obj. Ref. Node (now becomes/is replaced by said above nodes) when the New VI Obj. Ref. Node executes (FIG. 25A, Section 2 and/or FIG. 25C, Section 8) the code executes to provide the functionality of a "waveform chart UI node" (FIG. 25A, Section 2) or a "wait function node" (FIG. 25C, Section 8).

Cited Figure 4 (annotated), at least blocks 208-210 explicitly teaches "when the node in the graphical program executes, the determined graphical program code executes to provide the functionality in accordance with the selected function" (claim 69, lines 12-14):

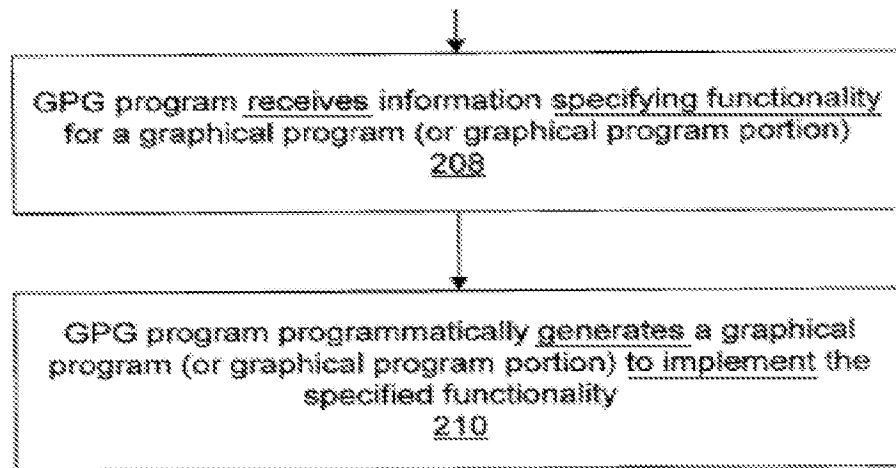


FIG. 4

A New VI Object Reference Node is illustrated in the annotated FIG. 13 below:

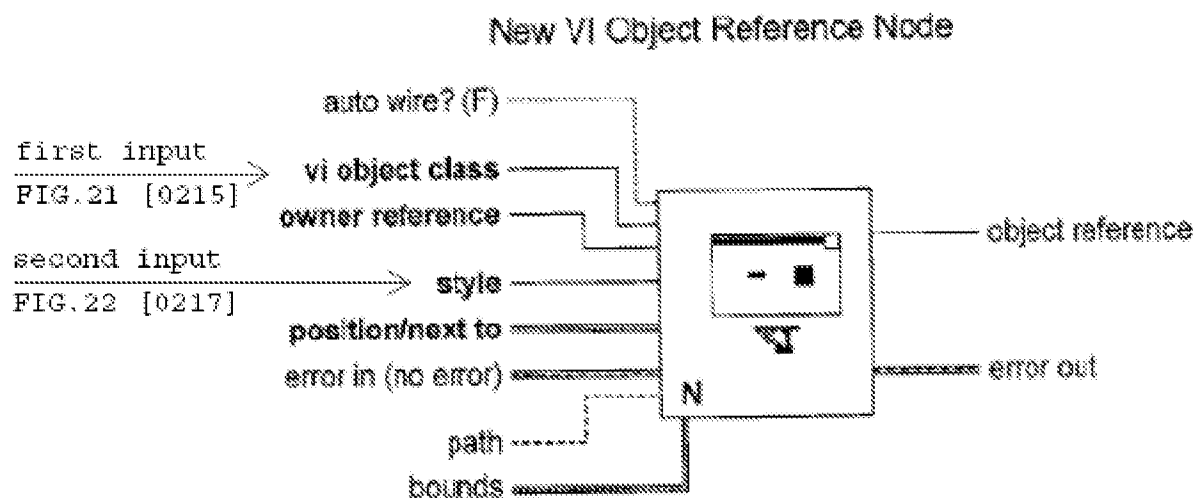


FIG. 13

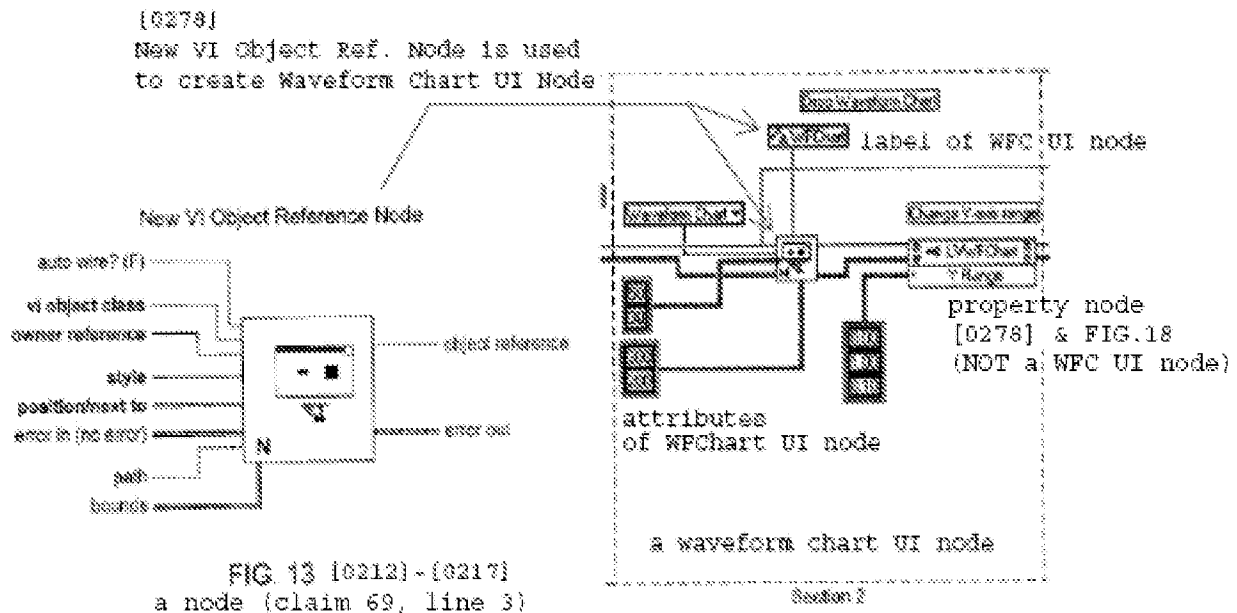
FIG. 25A, Section 2 further discloses:

“[0278] In section 2 of FIG. 25, a New VI Object Reference node is used to create the waveform chart user interface control (i.e., is used to create a selected function). As shown, the reference to the new graphical program generated by the New VI Reference node in section 1 is connected as the "owner



reference" input of the New VI Object Reference node. Other inputs are also connected which specify the type of object to create, the position to place the object, etc..." (i.e., after a user specifying functionality (FIG. 2, step 208) such as specifying first input (FIG. 21, [0215]) and second input (FIG. 22, [0217]), the New VI Object Reference node becomes/is replaced by the waveform chart user interface node, emphasis added).

Please see details in annotated FIG. 13 and FIG. 25A, Section 2, where the step of associating code with the node (FIG. 13, New VI Obj. Ref. Node) takes place:

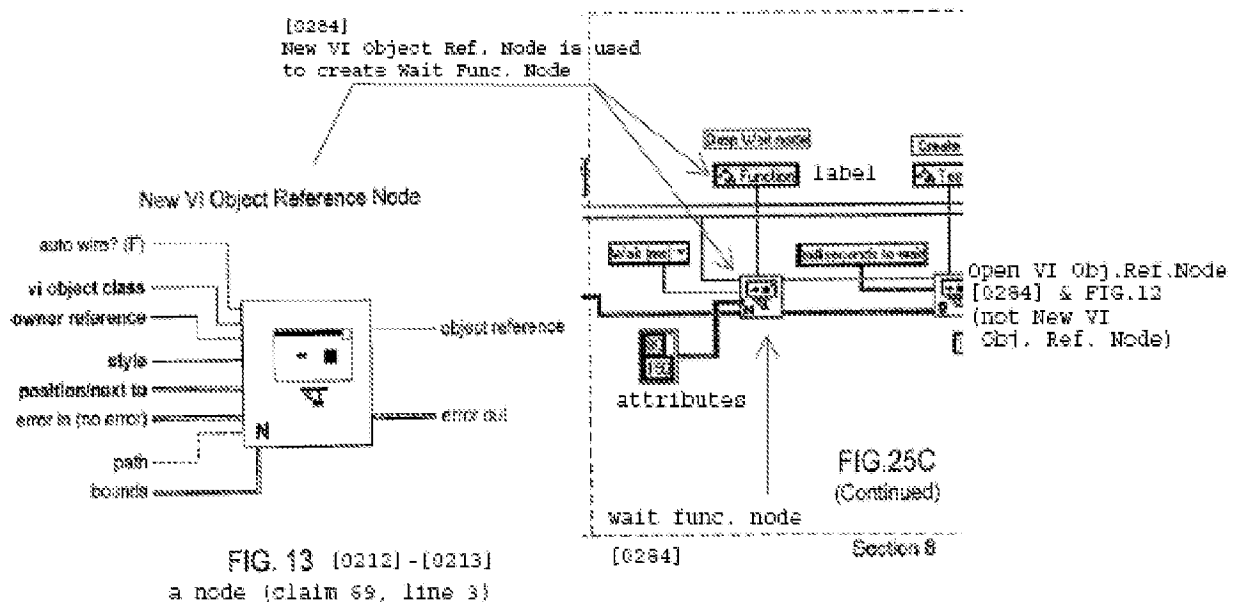


That is to say, the newly generated graphical program code (FIG. 4, step 210 above) implements (is connected/associated with) the waveform chart UI node (replacing the New VI Object Reference Node), so that when the New VI Object Reference Node (now becomes/is replaced by the waveform char UI node) executes, the newly generated graphical program code executes (FIG. 25A, Section 2) to provide

the functionality in accordance with the selected function (the waveform char UI function) also in FIG. 25A, Section 2 – emphasis added.

Similarly, Kudukoli also discloses such an “associating” feature that can be seen in annotated FIG. 13 and FIG. 25C, Section 8, for another selected/specified functionality as a wait function node - see [0284]:

“[0284] In section 8 of FIG. 25, a New VI Object Reference node is used to create a wait function node, an Open VI Object Reference node is used to obtain a reference to the "milliseconds to wait" input terminal of the wait function node ..." (i.e., after a user specifying functionality (FIG. 2, step 208) such as specifying first input (FIG. 21, [0215]) and second input (FIG. 22, [0217]), the New VI Object Reference node becomes/is replaced by the wait function node, emphasis added).



That is to say, the newly generated graphical program code (FIG. 4, step 210 above) implements (is connected/associated with) the wait function node (replacing the New VI Object Reference Node), so that when the New VI Object Reference Node (now

Art Unit: 2192

becomes/is replaced by the wait function node) executes, the newly generated graphical program code executes (FIG. 25C, Section 8) to provide the functionality in accordance with the selected function (the wait function) – emphasis added.

Accordingly, Appellant's general assertions are not persuasive. Kudukoli explicitly teaches “associating the determined graphical program code with the node, wherein, when the node in the graphical program executes, the determined graphical program code executes to provide the functionality in accordance with the selected function” as recited in independent claim 69, lines 12-14 and similarly recited in independent claim 77, lines 12-14.

**Dependent Claims 70-71 and 78-79 (Brief, pp. 9-10):**

The limitations at issue “wherein the node has a first node icon which is displayed in the graphical program, and wherein the first node icon has a first appearance, wherein the program instructions are further executable to perform: changing the first node icon to a second appearance based on the second user input, wherein said changing the first node icon to a second appearance includes displaying an image corresponding to the selected function”.

The examiner respectfully disagrees with Appellant's assertions. As illustrated in annotated FIG. 13 and FIG. 25A-C above, the New VI Object Reference Node has a first appearance in FIG. 13, and later has a second appearance associated with

either a waveform chart UI node (FIG. 25A, Section 2, waveform chart UI node with new/second appearance/attributes such as “owner reference”, “type of object”, “position”, see [0278]);

or a wait function node (FIG. 25C, Section 8, wait function node with new/second appearance/attributes such as “Wait (ms)”, see [0284]).

Art Unit: 2192

**Dependent Claims 72 and 80** (Brief, page 10):

The limitations at issue "prior to said associating the determined graphical program code with the node, the node does not have any associated graphical program code" (emphasis added).

The examiner respectfully disagrees with Appellant's assertions. The Appellant did not argue about the applied text portions (e.g., [0212]-[0221]) in the previous Office action mailed March 10, 2008, page 4.

Kudukoli explicitly teaches the New VI Object Reference Node does not have any associated graphical program node before receiving first and second user input (e.g., FIG. 13, [0212]-[0221]) – emphasis added.

**Dependent Claims 73 and 81** (Brief, pp. 11-12):

The limitations at issue "wherein, prior to said associating the determined graphical program code with the node, the node has associated default graphical program code in accordance with a default function for the node, and wherein the default graphical program code implements a first functionality; and wherein said associating the determined graphical program code with the node comprises replacing the default graphical program code with the determined graphical program code".

The examiner respectfully disagrees with Appellant's assertions. Kudukoli explicitly teaches "upcasting" and/or "downcasting" in [0222]-[0230], which may cast the type of a general slide object (i.e., a superclass with a default graphical program code) to the type of a vertical pointer slide object (i.e., replacing the superclass with a default graphical program code with the graphical program code of the vertical pointer slide object).

**Dependent Claims 74 and 82** (Brief, page 12):

The limitations at issue “wherein said receiving first user input comprises receiving the first user input to the node; and wherein said receiving second user input comprises receiving the second user input to the node”.

The examiner respectfully disagrees with Appellant’s assertions. As set forth in the previous Office action, pages 3 and 5, Kudukoli explicitly teaches:

*receiving first user input invoking display of a plurality of functions for the node; displaying the plurality of functions for the node in response to the first user input; (e.g., FIG. 21, from said new VI Object Node, invoking display of a plurality of functions such as “Numeric”, “Digital”, “Slide”, ..., [0215-0216]; FIG. 25A, section 2, [0278], from said new VI Object Node, invoking display of functions “Control”, “User Control”, “ActiveX” ..., see also FIG. 22);*

*receiving second user input selecting a function from the plurality of functions (e.g., FIG. 22, from “Slide”, selecting a specific function “Vertical Pointer Slide”, [0217]; from “User Control”, selecting a specific function “Waveform Chart”, see also [0278] and FIG. 22).*

**Dependent Claims 75 and 83 (Brief, pp. 12-13):**

The limitations at issue “said displaying the plurality of functions for the node in response to the first user input comprises: displaying a plurality of function classes for the node; and in response to user input selecting a function class, displaying the plurality of functions, wherein the plurality of functions are in the selected function class”.

The examiner respectfully disagrees with Appellant’s assertions. As set forth above, Kudukoli explicitly teaches a plurality of function classes such as general slide object (superclass) and vertical pointer slide object (sub-class) at least in paragraphs [0222]-[0230]. Kudukoli further teaches:

*said displaying the plurality of functions for the node in response to the first user input comprises: displaying a plurality of function classes for the node (e.g., FIG. 25A, section 2, [0278]); and*

*in response to user input selecting a function class, displaying the plurality of functions, wherein the plurality of functions are in the selected function class (e.g., [0217], [0278]).*

**Dependent Claims 76 and 84** (Brief, pp. 13-15):

The limitations at issue “a data acquisition (DAQ) node..., the DAQ node comprises one of: ...; a generic timing node; ..., after said associating, the DAQ node comprises one of: ...; a specific timing node in accordance with the selected function..”.

The examiner respectfully disagrees with Appellant’s assertions. Kudukoli explicitly teaches a DAQ node in LabView graphical programming environment ([0015] and [0085]), the DAQ node comprises a generic timing node such as a wait node (FIG. 23, [0275]), and after said associating, said DAQ node comprises a specific timing node with specific input/attributes (FIG. 25C, Section 8, a specific wait node must wait 100 milliseconds and connected to a Invoke node, [0284]).

**Independent Claims 85 and 89** (Brief, pp. 15-18):

The limitations at issue “determining a second node based on the selected function, wherein the second node comprises a graphical representation of an implementation of the selected function, and wherein the second node comprises graphical program code executable to provide functionality in accordance with the selected function” (Brief, pp. 15-16).

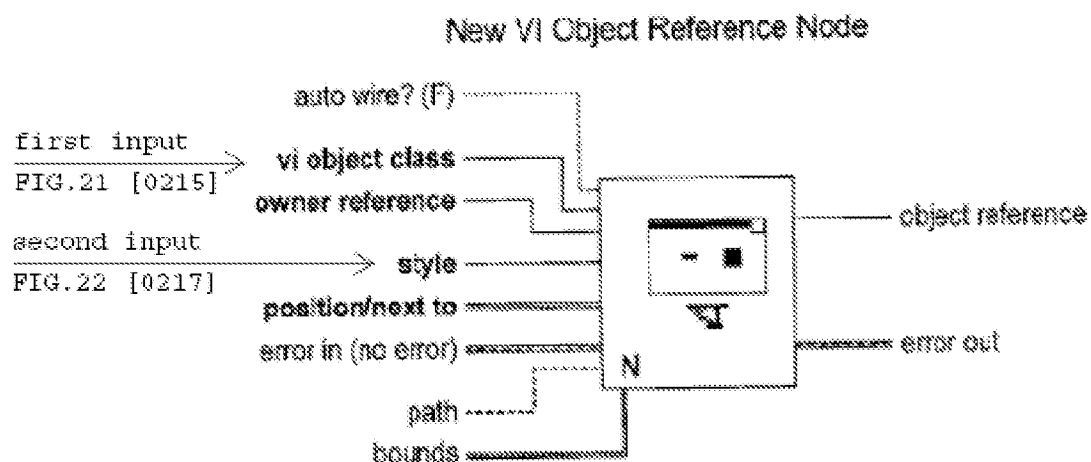
The examiner respectfully disagrees with Appellant’s assertions. Kudukoli explicitly teaches:

“[0212] FIG. 13--New VI Object Reference Node

[0213] FIG. 13 illustrates the New VI Object Reference node. The New VI Object Reference node creates a new VI object and outputs a reference to the new VI object. The following describes the inputs and outputs of the New VI Object Reference node: .....

[0215] vi object class specifies the type of object to create. Both objects for a front panel (user interface panel) and for a block diagram may be created. FIG. 21 illustrates how a user may choose a value for the vi object class input by selecting from a hierarchical menu. For example, a "slide" value may be chosen to designate that the reference to obtain is a reference to a slide user interface control. ....

[0217] style specifies the style or sub-class of object to create. FIG. 22 illustrates how a user may choose a value for the style input by selecting from a hierarchical menu. For example, if "slide" is chosen as the vi object class input, then "vertical pointer slide" may be chosen for the style input. The style input is ignored if the path input is wired".



**FIG. 13**

Based on the first input “vi object class” (FIG. 21, [0215]) and/or second input “style” (FIG. 22, [0217]), a New VI Object Reference Node is used to create a vertical pointer slide node ([0215] and [0217]), a waveform chart UI node (FIG. 25A, Section 2), or a wait function node (FIG. 25C, Section 8) as set forth in response to Claims 69 and 77 above.

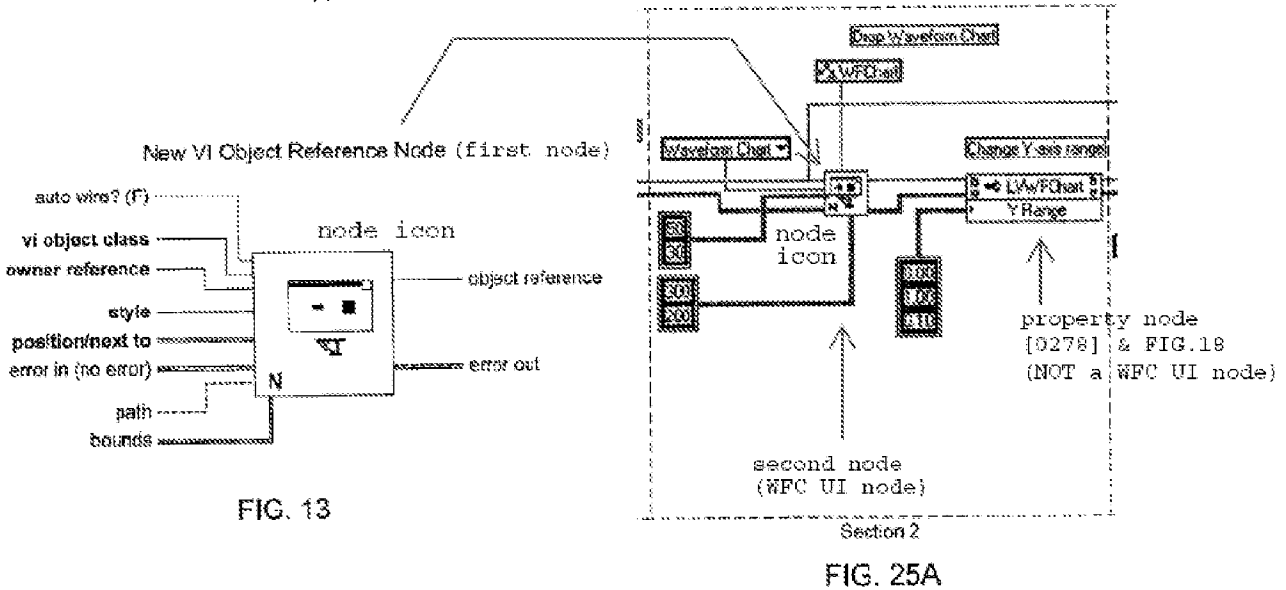
**Other limitations at issue** “replacing the node in the graphical program with the second node, wherein, when the second node in the graphical program executes, the determined graphical program code executes to provide the functionality in accordance with the selected function” (Brief, pp. 17-18, emphasis added).

The examiner respectfully disagrees with Appellant’s assertions. In dependent claim 86, the Appellant further defined “wherein the node comprises a first node icon, and wherein said displaying the node comprises displaying the first node icon, and wherein the second node comprises: the first node icon and the graphical program code; or a second node icon and the graphical program code” (emphasis added).

That is to say, the second node (either the waveform chart UI node in FIG. 25A, Section 2 or the wait function node in FIG. 25C, Section 8) has the first node icon of the first node (the node icon of New VI Object Reference Node) and the newly created graphical program code (used/connected/associated with either the waveform chart UI node or the wait function node) as follows:

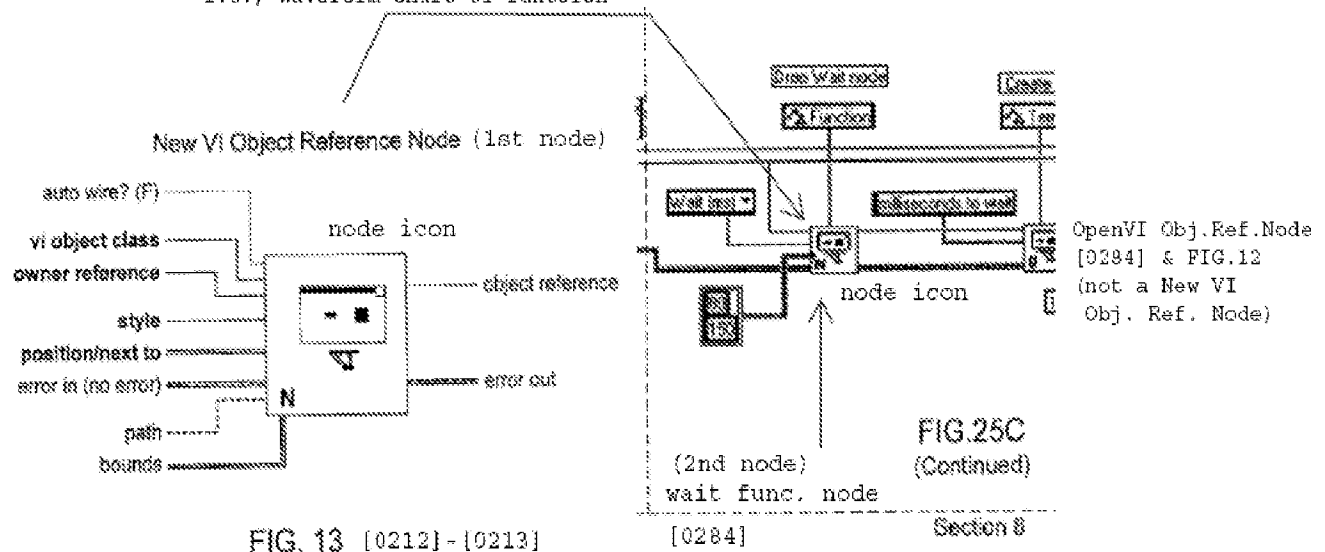


Waveform Chart UI Node (2nd node)  
has the same node icon of  
the New VI Object Ref. Node (1st node) and/plus  
the newly generated graphical program code,  
i.e., Waveform Chart UI function



and further in annotated FIG. 13 and FIG. 25C, Section 8:

Wait Func. Node (2nd node)  
has the same node icon of  
the New VI Object Ref. Node (1st node) and plus  
the newly generated graphical program code  
i.e., Waveform Chart UI function !



Art Unit: 2192

**Dependent Claims 86 and 90** (Brief, pp. 18-19):

Claims 86 and 90 recite the same limitations, which are incorporated into claims 85 and 89, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claims, it also teaches all of the limitations of claims 86 and 90.

**Dependent Claims 87 and 91** (Brief, page 19):

The limitations at issue “the node and/or the second node is one or more of: polymorphic; function switchable; or function class switchable”.

The examiner respectfully disagrees with Appellant’s assertions. Kudukoli explicitly teaches a New VI Object Reference Node (FIG. 13) may be replaced by either a waveform chart UI node (FIG. 25A, Section 2) or a wait function node (FIG. 25C, Section 8), which clearly the New VI Object Reference Node is polymorphic.

**Dependent Claims 88 and 92** (Brief, pp. 19-21):

Claims 86 and 90 recite the same limitations in claims 86 and 84, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claims, it also teaches all of the limitations of claims 88 and 92.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner’s answer.

Art Unit: 2192

For the above reasons, it is believed that the rejection should be sustained.

Respectfully submitted,

/Thuy Dao/

Examiner, Art Unit 2192

Conferees:

/Tuan Q. Dam/

Tuan Q. Dam

Supervisory Patent Examiner, Art Unit 2192

/Wendy Garber/

Director, Technology Center 2100